Project Documentation

1. Cover Page:

a. Project Title: Creating a Virtual Mouse using Python

b. Name: [Your Name]

Enrol. No.: [Your Enrolment Number]

Admission No.: [Your Admission Number]

Course: [Course Name], Group: [Group Name]

c. Faculty Name: [Faculty Name]

1. Table of Contents:
   1. Project Description
   2. Use-Case Diagram
   3. API Documentation
   4. Code Documentation
   5. Coding Conventions
   6. Citations
2. Project Description: a. Detailed Description: The project aims to create a virtual mouse using Python. The virtual mouse will work as a substitute for a physical mouse and can be used to perform various mouse-related actions such as clicking, dragging, and scrolling. b. Scope of the Project: The virtual mouse can be used by people who have difficulty using a physical mouse or for people who want to use a different type of mouse. The project can also be extended to create a virtual keyboard or other input devices. c. Details about the Stakeholders: The stakeholders for the project are people who have difficulty using a physical mouse or for people who want to use a different type of mouse. d. Brief Survey on Existing Techniques: There are various existing techniques for creating a virtual mouse such as using a camera-based approach or using an accelerometer-based approach.
3. Use-Case Diagram: The use-case diagram for the virtual mouse is as follows: [Insert image of use-case diagram]
4. API Documentation: a. Brief Introduction about Python: Python is a high-level, interpreted programming language that is used for a wide range of purposes such as web development, data analysis, artificial intelligence, and machine learning. It is easy to learn and has a simple syntax. b. Packages Required for the Project: The packages required for the project are: i. PyAutoGUI ii. OpenCV iii. NumPy c. Introduction to the Installation of Packages: i. PyAutoGUI: To install PyAutoGUI, open the terminal/command prompt and run the command "pip install pyautogui". ii. OpenCV: To install OpenCV, open the terminal/command prompt and run the command "pip install opencv-python". iii. NumPy: To install NumPy, open the terminal/command prompt and run the command "pip install numpy".
5. Code Documentation: The code for the virtual mouse is as follows:

FULL CODE :

import cv2

import mediapipe as mp

import pyautogui

cap = cv2.VideoCapture(0)

hand\_detector = mp.solutions.hands.Hands()

drawing\_utils = mp.solutions.drawing\_utils

screen\_width, screen\_height = pyautogui.size()

index\_y = 0

while True:

    \_, frame = cap.read()

    frame = cv2.flip(frame, 1)

    frame\_height, frame\_width, \_ = frame.shape

    rgb\_frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

    output = hand\_detector.process(rgb\_frame)

    hands = output.multi\_hand\_landmarks

    if hands:

        for hand in hands:

            drawing\_utils.draw\_landmarks(frame, hand)

            landmarks = hand.landmark

            for id, landmark in enumerate(landmarks):

                x = int(landmark.x\*frame\_width)

                y = int(landmark.y\*frame\_height)

                if id == 8:

                    cv2.circle(img=frame, center=(x,y), radius=10, color=(0, 255, 255))

                    index\_x = screen\_width/frame\_width\*x

                    index\_y = screen\_height/frame\_height\*y

                if id == 4:

                    cv2.circle(img=frame, center=(x,y), radius=10, color=(0, 255, 255))

                    thumb\_x = screen\_width/frame\_width\*x

                    thumb\_y = screen\_height/frame\_height\*y

                    print('outside', abs(index\_y - thumb\_y))

                    if abs(index\_y - thumb\_y) < 20:

                        pyautogui.click()

                        pyautogui.sleep(1)

                    elif abs(index\_y - thumb\_y) < 100:

                        pyautogui.moveTo(index\_x, index\_y)

    cv2.imshow('Virtual Mouse', frame)

    cv2.waitKey(1)

EXPLAINING THE CODE IN 5 PARTS :

Step 1 : OPENING UP THE VIDEO CAMERA

import cv2

cap = cv2.VideoCapture(0)

while True:

    \_, frame = cap.read()

    cv2.imshow('Virtual Mouse', frame)

    cv2.waitKey(1)

Step 2 :

1. Coding Conventions: The coding conventions used for the project are as follows: a. Variable names should be in lowercase and words should be separated by an underscore. b. Constants should be in uppercase and words should be separated by an underscore. c. Function names should be in lowercase and words should be separated by an underscore. d. Indentation should be done using four spaces. e. Comments should be used to explain the purpose of the code.
2. Citations: The following sources were used in the creation of the virtual mouse project: [Insert sources here]